jinyPHP 3권 – 08. 정규표현식

**08. 정규표현식**

===================

PHP는 정규표현에 관련된 기능과 함수를 제공합니다. 정규표현식은 PHP언어 이외에도 다른 언어에서 일반적으로 사용하는 패턴을 처리하는 개별 언어일종 입니다.

PHP는 정규표현에 대해서 perl(PCRE) 방식을 지원 합니다. 기존 POSIX 스타일의 정규표현은 PHP 5.3.x 이후에서 삭제되었습니다.

정규표현은 특수한 문자들의 패턴입니다.

**08.1 정규패턴**

===================

내부함수 preg\_match()는 패턴 정규표현식을 입력된 문자열 $subject에서 찾습니다. preg\_match() 함수는 subject 문자열에서 패턴을 매칭하여 결과값을 $matches 변수에 배열로 반환을 합니다.

**|내부함수|**

|  |
| --- |
| int **preg\_match** ( string $pattern , string $subject [, array &$matches [, int $flags = 0 [, int $offset = 0 ]]] ) |

이전에 ereg() 함수를 사용을 하였으나 PHP 7.x 에서 예전함수는 삭제되었습니다.

* matches: 세번째 인자 matches가 있을 경우에는 검색 결과를 배열로 반환을 합니다.
* flags : PREG\_OFFSET\_CAPTURE 값은 매치된 모든 문자열의 시작위치를 같이 반환합니다.
* offset: 문자열 검색의 시작 포인트를 지정할 수 있습니다. 기본값은 0 입니다.

**예제) preg-01.php**

|  |
| --- |
| <?php  $string = "2017-06-33";  $pattern = "/([0-9]{4})-([0-9]{1,2})-([0-9]{1,2})/";  if ( preg\_match($pattern, $string,$matches) ) {  echo"날짜 형식이 맞습니다.<br>";  if (checkdate($match[2],$match[3],$match[1])) {  echo "유효한 날짜입니다.<br>";  } else {  echo "날짜 형식은 맞지만, 유효한 날짜는 아닙니다.<br>";  }  } else {  echo"날짜 형식이 다릅니다.<br>";  }  ?> |

**화면출력)**

날짜 형식이 맞습니다.

날짜 형식은 맞지만, 유효한 날짜는 아닙니다.

**08.2 패턴변환**

===================

내부함수 preg\_replace()는 패턴을 찾아서 다음 문자열로 변환을 합니다. 문자열 subject 에서 패턴 배열의 값과 같은 값들을 검사하여 replacement 문자열로 변환을 합니다.

**|내부함수|**

|  |
| --- |
| mixed **preg\_replace** ( mixed $pattern , mixed $replacement , mixed $subject [, int $limit [, int &$count ]] ) |

**예제) preg-02.php**

|  |
| --- |
| <?php  $string = 'jun 21, 2017';  $pattern = '/(\w+) (\d+), (\d+)/i';  $replacement = '${1} ${2}, $3';  echo preg\_replace($pattern, $replacement, $string);  echo "<br>";  $string = 'hello world! my name is jiny.';    $patterns[0] = '/hello world/';  $patterns[1] = '/my name is/';  $patterns[2] = '/jiny/';  $replacements[2] = '안녕하세요';  $replacements[1] = '제 이름은 ';  $replacements[0] = '지니';  echo preg\_replace($patterns, $replacements, $string);  ?> |

**화면출력)**

jun 21, 2017

안녕하세요! 제 이름은 지니.

**08.3 패턴분리**

===================

내부함수 preg\_split()는 정규표현 패턴에 따라서 문자열을 분리하여 배열로 반환을 합니다.

**|내부함수|**

|  |
| --- |
| rray **preg\_split** ( string $pattern , string $subject [, int $limit [, int $flags ]] ) |

**예제) preg-03.php**

|  |
| --- |
| <?php  // " ", \r, \t, \n, \f를 포함하여  $keywords = preg\_split("/[\s,]+/", "php language, programming");  echo "임의 갯수의 콤마와 스페이스로 구문을 나눕니다. ";  print\_r($keywords);  echo "<br>";  $str = 'string';  echo "문자 단위로 분리합니다. ";  $chars = preg\_split('//', $str, -1, PREG\_SPLIT\_NO\_EMPTY);  print\_r($chars);  echo "<br>";  ?> |

**화면출력)**

임의 갯수의 콤마와 스페이스로 구문을 나눕니다. Array ( [0] => php [1] => language [2] => programming )

문자 단위로 분리합니다. Array ( [0] => s [1] => t [2] => r [3] => i [4] => n [5] => g )

**|내부함수|**

|  |
| --- |
| string **preg\_quote** ( string $str [, string $delimiter ] ) |

내부함수 preg\_quote()는 정규표현식의 문자를 인용합니다. preg\_quote() 함수는 입력된 문자열에 대해서 모든 문자앞에 백슬래시를 추가합니다.

이러한 표현은 정규표현식을 처리하는 문자열 처리할때 유용합니다.

정규 표현식 특수 문자는: . \ + \* ? [ ^ ] $ ( ) { } = ! < > | :

**예제) preg-04.php**

|  |
| --- |
| <?php  $keywords = '$123 for a hojin/jiny';  $keywords = preg\_quote($keywords, '/');  echo $keywords;  echo "<br>";  // preg\_quote($word)는 정규 표현식에서 애스터라이크(\*) 처리  $body = "안녕하세요 본 책은 매우 \*쉽게\* 작성을 하였습니다.";  $word = "\*매우\*";  $body = preg\_replace("/".preg\_quote($word)."/", "<i>" . $word . "</i>", $body);  echo $body;  ?> |

**화면출력)**

\$123 for a hojin\/jiny

안녕하세요 본 책은 매우 \*쉽게\* 작성을 하였습니다.